The Claims

What is claimed as the invention is:

1. In a computer network having at least one client and at least one server, said client and said server being selectively in communication with each other over said network, said server streaming into said network a plurality of RTP packets addressed for said client at a normal streaming rate commensurate with a rate of reading said packets at said client, each of said RTP packets including at least a sequence number and a timestamp, a reliable RTP method comprising:

acknowledging to said server each of said packets received by said client;

re-transmitting from said server to said client any of said packets that remain unacknowledged subsequent to expiration of a predetermined time duration subsequent to said timestamp;

continuously determining a maximum number of bytes that may be contained in said RTP packets streaming into said network and, in the event a number of bytes in said RTP packets exceeds said maximum number, discontinuing streaming of said RTP packets until said determining step indicates said number of bytes is less than said maximum number; and

continuously determining a present streaming rate at which said RTP packets are streamed into said network wherein said present streaming rate exceeds said normal streaming rate.

2. A method as set forth in Claim 1 wherein said acknowledging includes sending from said client to said server of plurality of ACK packets in response to receiving said RTP packets.

3. A method as set forth in Claim 2 wherein said sending step includes inserting into each of said ACK packets said sequence number of at least a respective one of said RTP packets received at said client.

4. A method as set forth in Claim 3 wherein said inserting step further includes inserting a bit mask representing an offset from said sequence number into each of ACK packets.

5. A method as set forth in Claim 2 further comprising:

computing said predetermined time duration as an estimated round-trip time.

6. A method as set forth in Claim 5 wherein said computing step includes:

measuring a time period from transmission of each one of said RTP packets streamed by said server to receipt by said server of said ACK packets acknowledging each respective one of said RTP packets.

7. A method as set forth in Claim 6 wherein said measuring step includes:

marking a time of transmission for each one of said RTP packets streamed from said server;

marking a time of receipt for said ACK packets acknowledging each respective one of said RTP packets;

calculating as a function of said time of transmission and said time of receipt said estimated round-trip time.

8. A method as set forth in Claim 7 wherein said calculating step utilizes Karn's algorithm.

9. A method as set forth in Claim 6 wherein said computing step includes ignoring said time period for any one of said RTP packets having been re-transmitted prior to receipt of by said server of one of said ACK packets acknowledging said any one of said RTP packets.

10. A method as set forth in Claim 5 further comprising:

initializing a minimum round-trip threshold.

11. A method as set forth in Claim 10 further comprising:

resetting said minimum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is less than said minimum round-trip threshold.

12. A method as set forth in Claim 5 further comprising:

initializing a maximum round-trip threshold.

13. A method as set forth in Claim 12 further comprising:

resetting said maximum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is greater than said maximum round-trip threshold.

14. A method as set forth in Claim 12 wherein said initializing step includes initializing said maximum round-trip threshold equal to an initial minimum round-trip threshold.

15. A method as set forth in Claim 5 further comprising increasing said estimated round-trip time upon an occurrence of said re-transmitting step.

16. A method as set forth in Claim 15 wherein said increasing step includes multiplying said estimated round-trip time by a predetermined coefficient.

17. A method as set forth in Claim 16 wherein said predetermined coefficient is equal to 3/2.

18. A method as set forth in Claim 2 wherein said maximum number of bytes determining step includes:

computing a congestion window size; and

computing a difference between a number of bytes in said RTP packets currently streamed into said network and a number of bytes in said RTP packets acknowledged by said ACK packets, said maximum number of bytes being a number of bytes by which said congestion window size exceeds said difference.

19. A method as set forth in Claim 18 wherein said congestion window size computing step includes:

setting said congestion window size to an initial congestion window size;

varying said congestion window size constrained by a maximum congestion window size in response to receiving said ACK packets.

20. A method as set forth in Claim 19 wherein said setting step includes computing said initial congestion window size as a selected multiple of a maximum segment size.

21. A method as set forth in Claim 20 wherein said selected multiple of said maximum segment size is four.

22. A method as set forth in Claim 19 wherein said varying step includes functionally computing said congestion window size as a function of a selected one of a maximum segment size and a number of bytes in each of said RTP packets for which a respective one of said ACK packets has been received.

23. A method as set forth in Claim 22 wherein said functionally computing step includes increasing said congestion window size by a number of bytes in each one of said RTP packets for which a respective one of said ACK packets has been received.

24. A method as set forth in Claim 23 wherein said increasing step is performed only in the event of said congestion window size is presently below a slow start threshold.

25. A method as set forth in Claim 22 wherein said functionally computing step includes increasing said congestion window size for each full window of said ACK packets received.

26. A method as set forth in Claim 25 wherein said increasing step is performed only in the event said congestion window size is presently above a slow start threshold.

27. A method as set forth in Claim 19 wherein said congestion window size computing step further includes resetting said congestion window size to a lesser of one-half of a slow start threshold and one-half of a current congestion window size upon an occurrence of said retransmitting step.

28. A method as set forth in Claim 19 wherein said varying step includes setting said maximum congestion window size equal to a size of a client window.

29. A method as set forth in Claim 1 wherein said re-transmitting step includes inserting into each of said RTP packets an expiration time, said any of said packets remaining unacknowledged not being re-transmitted in the event said expiration time is less than said predetermined time duration.

30. A method as set forth in Claim 1 wherein said maximum number of bytes determining step includes adding to said maximum number a number of bytes of said any of said RTP packets remaining unacknowledged after expiration of said time duration.

31. A method as set forth in Claim 1 wherein said streaming rate determining step includes:

reporting by said client to said server an overbuffer window size; and

setting said streaming rate at a rate above said rate of reading wherein said overbuffer window size is transmitted.

32. A method as set forth in Claim 31 wherein said reporting step includes:

inserting into an APP packet said overbuffer window size; and

sending by said client to said server said APP packet.

33. A method as set forth in Claim 31 wherein said streaming rate determining step further includes discontinuing streaming of said RTP packets when said overbuffer window is full.

34. A method as set forth in Claim 1 further comprising:

inserting into a setup request first header information communicated to said server by said client to initiate said reliable RTP method; and

inserting said first header information identically into a setup response to be communicated to said client by said server.

35. A method as set forth in Claim 34 wherein said first header information includes a protocol name and at least one parameter following said protocol name.

36. A method as set forth in Claim 35 wherein said parameter includes a client window size.

37. A method as set forth in Claim 34 further comprising:

inserting into said setup request second header information indicative of transport options, said second header information being inserted identically into said setup response.

38. A method as set forth in Claim 37 wherein one of said options is a late tolerance option.

39. A computer system comprising:

a computer network;

a client including a buffer; and

a server, said client and said server being selectively in communication with each other over said network, said buffer temporarily storing a plurality of RTP packets streamed into said network by said server at a normal streaming rate commensurate with a rate of reading said packets by said client from said buffer, each of said RTP packets including at least a sequence number and a timestamp, said client acknowledging to said server each of said packets received by said client, said server

re-transmitting to said client any of said packets that remain unacknowledged subsequent to expiration of a predetermined time duration subsequent to said timestamp, said server continuously determining a maximum number of bytes that may be contained in said RTP packets streaming into said network and, in the event a number of bytes in said RTP packets exceeds said maximum number, discontinuing streaming of said RTP packets until said number of bytes is less than said maximum number, and said server further continuously determining a present streaming rate at which said RTP packets are streamed into said network wherein said present streaming rate exceeds said normal streaming rate.

40. A system as set forth in Claim 39 wherein said client develops a plurality of ACK packets in response to receiving said RTP packets, each of said ACK packet being sent from said client to said server to acknowledge receipt of said RTP packets.

41. A system as set forth in Claim 40 wherein each of said ACK packets includes said sequence number of at least a respective one of said RTP packets received at said client.

42. A system as set forth in Claim 41 wherein each of said ACK packets further includes a bit mask representing an offset from said sequence number.

43. A system as set forth in Claim 42 wherein said server further computes said predetermined time duration as an estimated round-trip time.

44. A system as set forth in Claim 43 wherein said server to compute said predetermined time duration further measures a time period from transmission of each

one of said RTP packets streamed by said server to receipt by said server of said ACK packets acknowledging each respective one of said RTP packets.

45. A system as set forth in Claim 44 wherein said server to measure said time period marks each of a time of transmission for each one of said RTP packets streamed from said server and a time of receipt for said ACK packets acknowledging each respective one of said RTP packets, and further wherein said server further calculates as a function of said time of transmission and said time of receipt said estimated round-trip time.

46. A system as set forth in Claim 45 wherein said function is Karn's algorithm.

47. A system as set forth in Claim 44 wherein said server ignores said time period for any one of said RTP packets having been re-transmitted prior to receipt of by said server of one of said ACK packets acknowledging said any one of said RTP packets.

48. A system as set forth in Claim 43 wherein said server further initializes a minimum round-trip threshold.

49. A system as set forth in Claim 48 wherein said server further resets said minimum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is less than said minimum round-trip threshold.

50. A system as set forth in Claim 43 wherein said server further initializes a maximum round-trip threshold.

51. A system as set forth in Claim 50 wherein said server further resets said maximum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is greater than said maximum round-trip threshold.

52. A system as set forth in Claim 50 wherein said server initializes said maximum round-trip threshold equal to an initial minimum round-trip threshold.

53. A system as set forth in Claim 43 wherein said server increases said estimated round-trip time upon an occurrence of said server re-transmitting said any of said RTP packets.

54. A system as set forth in Claim 53 wherein said estimated round-trip time is increased by a predetermined coefficient.

55. A system as set forth in Claim 54 wherein said predetermined coefficient is equal to 3/2.

56. A system as set forth in Claim 40 wherein said server to determine said maximum number of bytes computes a congestion window size and further computes a difference between a number of bytes in said RTP packets currently streamed into said network and a number of bytes in said RTP packets acknowledged by said ACK packets, said maximum number of bytes being a number of bytes by which said congestion window size exceeds said difference.

57. A system as set forth in Claim 56 wherein said server to compute said congestion window size sets said congestion window size to an initial congestion window size and further varies said congestion window size in response to receiving

said ACK packets, said congestion window size being constrained by a maximum congestion window size.

58. A system as set forth in Claim 57 wherein said initial congestion window size is a selected multiple of a maximum segment size.

59. A system as set forth in Claim 58 wherein said selected multiple is four.

60. A system as set forth in Claim 57 wherein said server to vary said congestion window size computes said congestion window size as a function of a selected one of a maximum segment size and a number of bytes in each of said RTP packets for which a respective one of said ACK packets has been received.

61. A system as set forth in Claim 60 wherein said function increases said congestion window size by a number of bytes in each one of said RTP packets for which a respective one of said ACK packets has been received.

62. A system as set forth in Claim 61 wherein said congestion window size is presently below a slow start threshold.

63. A system as set forth in Claim 60 wherein said function increases said congestion window size as a function of a square of said maximum segment size divided by a present size of said congestion window size for each one of said ACK packets received acknowledging said maximum segment size.

64. A system as set forth in Claim 63 wherein said congestion window size is presently above a slow start threshold.

65. A system as set forth in Claim 57 wherein said server resets said congestion window size to a lesser of one-half of a slow start threshold and one-half of a current congestion window size upon an occurrence of said server re-transmitting said any one of said RTP packets.

66. A system as set forth in Claim 57 wherein said maximum congestion window size is set equal to a size of a client window.

67. A system as set forth in Claim 39 wherein each of said RTP packets is associated with an expiration time, said any of said RTP packets remaining unacknowledged not being re-transmitted in the event said expiration time is less than said predetermined time duration.

68. A system as set forth in Claim 39 wherein said maximum number of bytes includes a number of bytes of said any of said RTP packets remaining unacknowledged after expiration of said time duration.

69. A system as set forth in Claim 39 wherein said client report to said server an overbuffer window size and said server in response thereto sets said streaming rate at a rate above said rate of reading wherein said overbuffer window size is transmitted.

70. A system as set forth in Claim 69 wherein said client develops an APP packet to report said overbuffer window size and further sends said APP packet to said server.

71. A system as set forth in Claim 69 wherein said server further discontinues streaming of said RTP packets when said overbuffer window is full.

72. A system as set forth in Claim 39 wherein said client to initiate communication over said network transmits a setup request to said server, said setup request including first header information, and said server in response thereto transmits a setup response to said client, said setup response including said first header information.

73. A system as set forth in Claim 72 wherein said first header information includes a protocol name and at least one parameter following said protocol name.

74. A system as set forth in Claim 73 wherein said parameter includes a client window size.

75. A system as set forth in Claim 72 wherein said setup request further includes second header information indicative of transport options, said second header information being inserted identically into said setup response.

76. A system as set forth in Claim 75 wherein one of said options is a late tolerance option.

77. In a computer network having at least one client and at least one server, said client and said server being selectively in communication with each other over said network, said server streaming into said network a plurality of RTP packets addressed for said client at a normal streaming rate commensurate with a rate of reading said packets at said client, each of said RTP packets including at least a sequence number and a timestamp, a computer readable medium containing programming code that when executed implements procedures comprising:

acknowledging to said server each of said packets received by said client;

re-transmitting from said server to said client any of said packets that remain unacknowledged subsequent to expiration of a predetermined time duration subsequent to said timestamp;

continuously determining a maximum number of bytes that may be contained in said RTP packets streaming into said network and, in the event a number of bytes in said RTP packets exceeds said maximum number, discontinuing streaming of said RTP packets until said determining procedure indicates said number of bytes is less than said maximum number; and

continuously determining a present streaming rate at which said RTP packets are streamed into said network wherein said present streaming rate exceeds said normal streaming rate.

78. A medium as set forth in Claim 77 wherein said acknowledging procedure includes sending from said client to said server of plurality of ACK packets in response to receiving said RTP packets.

79. A medium as set forth in Claim 78 wherein said sending procedure includes inserting into each of said ACK packets said sequence number of at least a respective one of said RTP packets received at said client.

80. A medium as set forth in Claim 79 wherein said inserting procedure further includes inserting a bit mask representing an offset from said sequence number into each of ACK packets.

81. A medium as set forth in Claim 78 further comprising procedures of: computing said predetermined time duration as an estimated round-trip time.

82. A medium as set forth in Claim 81 wherein said computing procedure includes procedures of:

measuring a time period from transmission of each one of said RTP packets streamed by said server to receipt by said server of said ACK packets acknowledging each respective one of said RTP packets.

83. A medium as set forth in Claim 82 wherein said measuring procedure includes procedures of:

marking a time of transmission for each one of said RTP packets streamed from said server;

marking a time of receipt for said ACK packets acknowledging each respective one of said RTP packets;

calculating as a function of said time of transmission and said time of receipt said estimated round-trip time.

84. A medium as set forth in Claim 83 wherein said calculating procedure utilizes Karn's algorithm.

85. A medium as set forth in Claim 82 wherein said computing procedure includes ignoring said time period for any one of said RTP packets having been re-transmitted prior to receipt of by said server of one of said ACK packets acknowledging said any one of said RTP packets.

86. A medium as set forth in Claim 81 further comprising procedures of:

initializing a minimum round-trip threshold.

87. A medium as set forth in Claim 86 further comprising procedures of:

resetting said minimum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is less than said minimum round-trip threshold.

88. A medium as set forth in Claim 81 further comprising procedures of:

initializing a maximum round-trip threshold.

89. A medium as set forth in Claim 86 further comprising procedures of:

resetting said maximum round-trip threshold to said estimated round-trip time in the event said estimated round-trip time is greater than said maximum round-trip threshold.

90. A medium as set forth in Claim 86 wherein said initializing procedure includes initializing said maximum round-trip threshold equal to an initial minimum round-trip threshold.

91. A medium as set forth in Claim 81 further comprising increasing said estimated round-trip time upon an occurrence of said re-transmitting procedure.

92. A medium as set forth in Claim 91 wherein said increasing procedure includes multiplying said estimated round-trip time by a predetermined coefficient.

93. A medium as set forth in Claim 92 wherein said predetermined coefficient is equal to 3/2.

94. A medium as set forth in Claim 78 wherein said maximum number of bytes determining procedure includes procedures of:

computing a congestion window size; and

computing a difference between a number of bytes in said RTP packets currently streamed into said network and a number of bytes in said RTP packets acknowledged by said ACK packets, said maximum number of bytes being a number of bytes by which said congestion window size exceeds said difference.

95. A medium as set forth in Claim 94 wherein said congestion window size computing procedure includes procedures of:

setting said congestion window size to an initial congestion window size;

varying said congestion window size constrained by a maximum congestion window size in response to receiving said ACK packets.

96. A medium as set forth in Claim 95 wherein said setting procedure includes computing said initial congestion window size as a selected multiple of a maximum segment size.

97. A medium as set forth in Claim 96 wherein said selected multiple of said maximum segment size is four.

98. A medium as set forth in Claim 95 wherein said varying procedure includes functionally computing said congestion window size as a function of a selected one of a maximum segment size and a number of bytes in each of said RTP packets for which a respective one of said ACK packets has been received.

99. A medium as set forth in Claim 98 wherein said functionally computing procedure includes increasing said congestion window size by a number of bytes in each one of said RTP packets for which a respective one of said ACK packets has been received.

100. A medium as set forth in Claim 99 wherein said increasing procedure is performed only in the event of said congestion window size is presently below a slow start threshold.

101. A medium as set forth in Claim 98 wherein said functionally computing procedure includes increasing said congestion window size for each full window of said ACK packets received.

102. A medium as set forth in Claim 101 wherein said increasing procedure is performed only in the event said congestion window size is presently above a slow start threshold.

103. A medium as set forth in Claim 95 wherein said congestion window size computing procedure further includes resetting said congestion window size to a lesser of one-half of a slow start threshold and one-half of a current congestion window size upon an occurrence of said re-transmitting procedure.

104. A medium as set forth in Claim 95 wherein said varying procedure includes setting said maximum congestion window size equal to a size of a client window.

105. A medium as set forth in Claim 77 wherein said re-transmitting procedure includes inserting into each of said RTP packets an expiration time, said any of said packets remaining unacknowledged not being re-transmitted in the event said expiration time is less than said predetermined time duration.

106. A medium as set forth in Claim 77 wherein said maximum number of bytes determining procedure includes adding to said maximum number a number of

bytes of said any of said RTP packets remaining unacknowledged after expiration of said time duration.

107. A medium as set forth in Claim 77 wherein said streaming rate determining procedure includes procedures of:

reporting by said client to said server an overbuffer window size; and

setting said streaming rate at a rate above said rate of reading wherein said overbuffer window size is transmitted.

108. A medium as set forth in Claim 107 wherein said reporting procedure includes procedures of:

inserting into an APP packet said overbuffer window size; and

sending by said client to said server said APP packet.

109. A medium as set forth in Claim 107 wherein said streaming rate determining procedure further includes discontinuing streaming of said RTP packets when said overbuffer window is full.

110. A medium as set forth in Claim 77 further comprising procedures of:

inserting into a setup request first header information communicated to said server by said client to initiate a connection between said client and said server; and

inserting said first header information identically into a setup response to be communicated to said client by said server.

111. A medium as set forth in Claim 110 wherein said first header information includes a protocol name and at least one parameter following said protocol name.

112. A medium as set forth in Claim 111 wherein said parameter includes a client window size.

113. A medium as set forth in Claim 110 further comprising procedures of:

inserting into said setup request second header information indicative of transport options, said second header information being inserted identically into said setup response.

114. A medium as set forth in Claim 113 wherein one of said options is a late tolerance option.

115. A computer system having at least one client, at least one server and a computer network, said client and said server being selectively in communication with each other over said network, said client having a buffer to store temporarily a plurality of packets streamed into said network by said server at a normal streaming rate commensurate with a rate of reading said packets by said client from said buffer, each of said RTP packets including at least a sequence number and a timestamp, said computer system comprising:

means for acknowledging to said server each of said packets received by said client;

means for re-transmitting from said server to said client any of said packets that remain unacknowledged subsequent to expiration of a predetermined time duration subsequent to said timestamp;

means for continuously determining a maximum number of bytes that may be contained in said RTP packets streaming into said network and, in the event a number of bytes in said RTP packets exceeds said maximum number, for discontinuing streaming of said RTP packets until said number of bytes is less than said maximum number; and

means for continuously determining a present streaming rate at which said RTP packets are streamed into said network wherein said present streaming rate exceeds said normal streaming rate.

116. A system as set forth in Claim 115 wherein said acknowledging means sends from said client to said server of plurality of ACK packets in response to receiving said RTP packets.

117. A system as set forth in Claim 116 wherein each of said ACK packets includes said sequence number of at least a respective one of said RTP packets received at said client.

118. A system as set forth in Claim 117 wherein each of said ACK packets further include a bit mask representing an offset from said sequence number into each of ACK packets.

119. A system as set forth in Claim 116 further comprising:

means for computing said predetermined time duration as an estimated round-trip time.

120. A system as set forth in Claim 119 wherein said computing means measures a time period from transmission of each one of said RTP packets streamed by said server to receipt by said server of said ACK packets acknowledging each respective one of said RTP packets.

121. A system as set forth in Claim 120 wherein said computing means to measure said time period marks a time of transmission for each one of said RTP packets streamed from said server and marks a time of receipt for said ACK packets

acknowledging each respective one of said RTP packets, said estimated round-trip time being calculated as a function of said time of transmission and said time of receipt.

122. A system as set forth in Claim 121 wherein said estimated round-trip time is calculated using Karn's algorithm.

123. A system as set forth in Claim 120 wherein said time period for any one of said RTP packets having been re-transmitted prior to receipt of by said server of one of said ACK packets acknowledging said any one of said RTP packets is ignored.

124. A system as set forth in Claim 119 further comprising:

means for initializing a minimum round-trip threshold.

125. A system as set forth in Claim 124 wherein said minimum round-trip threshold is reset to said estimated round-trip time in the event said estimated round-trip time is less than said minimum round-trip threshold.

126. A system as set forth in Claim 119 further comprising:

means for initializing a maximum round-trip threshold.

127. A system as set forth in Claim 126 wherein said maximum round-trip threshold is reset to said estimated round-trip time in the event said estimated round-trip time is greater than said maximum round-trip threshold.

128. A system as set forth in Claim 126 wherein said maximum round-trip threshold is initialized to equal to an initial minimum round-trip threshold.

129. A system as set forth in Claim 119 wherein said estimated round-trip time is increased in response to an occurrence of re-transmitting said any one of said RTP packets.

130. A system as set forth in Claim 129 wherein said estimated round-trip time is multiplied by a predetermined coefficient.

131. A system as set forth in Claim 130 wherein said predetermined coefficient is equal to 3/2.

132. A system as set forth in Claim 116 wherein said maximum number of bytes determining means further computes each of a congestion window size and a difference between a number of bytes in said RTP packets currently streamed into said network and a number of bytes in said RTP packets acknowledged by said ACK packets, said maximum number of bytes being a number of bytes by which said congestion window size exceeds said difference.

133. A system as set forth in Claim 132 wherein said congestion window size is set to an initial congestion window size and subsequently varied in response to receiving said ACK packets, said congestion window size being constrained by a maximum congestion window size.

134. A system as set forth in Claim 133 wherein said initial congestion window size is a selected multiple of a maximum segment size.

135. A system as set forth in Claim 134 wherein said selected multiple is four.

136. A system as set forth in Claim 133 wherein said congestion window size is varied as a function of a selected one of a maximum segment size and a number of bytes in each of said RTP packets for which a respective one of said ACK packets has been received.

137. A system as set forth in Claim 136 wherein said function increases said congestion window size by a number of bytes in each one of said RTP packets for which a respective one of said ACK packets has been received.

138. A system as set forth in Claim 137 wherein said function is operative only in the event of said congestion window size is presently below a slow start threshold.

139. A system as set forth in Claim 136 wherein said function increases said congestion window size for each full window of said ACK packets received.

140. A system as set forth in Claim 139 wherein said function is operative only in the event said congestion window size is presently above a slow start threshold.

141. A system as set forth in Claim 133 wherein said congestion window size is reset to a lesser of one-half of a slow start threshold and one-half of a current congestion window size upon an occurrence of said any of said RTP packets being re-transmitted.

142. A system as set forth in Claim 133 wherein said maximum congestion window size is equal to a size of a client window.

143. A system as set forth in Claim 115 wherein each of said RTP packets includes an expiration time, said any of said packets remaining unacknowledged not being re-transmitted in the event said expiration time is less than said predetermined time duration.

144. A system as set forth in Claim 115 wherein said maximum number of bytes includes a number of bytes of said any of said RTP packets remaining unacknowledged after expiration of said time duration.

145. A system as set forth in Claim 115 wherein said streaming rate determining means includes:

means reporting by said client to said server an overbuffer window size; and said server in response to said reporting means setting said streaming rate at a rate above said rate of reading wherein said overbuffer window size is transmitted.

146. A system as set forth in Claim 145 wherein said reporting means inserts into an APP packet said overbuffer window size, and sends said APP packet to said server.

147. A system as set forth in Claim 145 wherein said streaming rate determining means discontinues streaming of said RTP packets when said overbuffer window is full.

148. A system as set forth in Claim 115 further comprising:

means for sending a setup request communicated to said server by said client to initiate a connection over said network, said setup request including first header information; and

means for sending a setup response communicated to said client by said server, said setup response including said first header information.

149. A system as set forth in Claim 148 wherein said first header information includes a protocol name and at least one parameter following said protocol name.

150. A system as set forth in Claim 149 wherein said parameter includes a client window size.

151. A system as set forth in Claim 148 wherein said setup request includes second header information indicative of transport options, said second header information being inserted identically into said setup response.

152. A system as set forth in Claim 151 wherein one of said options is a late tolerance option.